Multilevel Timing-Constrained Full-Chip Routing in Hierarchical Quad-Grid Model

Abstract

Given a set of timing-driven routing trees for all the interconnection nets, a new multilevel timing-constrained full-chip routing(MTFR) in a dynamic hierarchical quad-grid model is proposed to complete full-chip routing in reasonable time. The experimental results show that the proposed MTFR approach uses less CPU time to obtain 100% timing-constrained routing results for all the tested benchmark circuits.

1. Introduction

Basically, the concept of multilevel designs has been successfully applied to VLSI physical design. For example, multilevel partitioners, *ML*[1], *hMETIS*[2], and *HPM*[3], multilevel placer, *mPL*[4] and multilevel floorplanner, *MB*-tree*[5]. It is assumed that multilevel technique is implemented in a two-stage process: coarsening and uncoarsening. The coarsening stage iteratively groups a set of circuit components until the number of considered components is smaller than a given threshold, and a given problem is solved in the final size. Furthermore, the uncoarsening stage iteratively ungroups a set of previously clustered circuit components and refines the solution by using an improvement method. As similar to the concept of multilevel routing, Lin Hsu and Tsai[6] presented a hybrid hierarchical approach with the bottom-up(coarsening) and top-down(uncoarsening) technique for global routing. Recently, Cong, Fang and Zhang[7] proposed a multilevel approach for large-scale full-chip routability-driven global routing. Again, Cong, Xie and Zhang[8] presented an enhanced multilevel routing system, named MARS, which incorporates several new techniques of resource reservation for local nets during the coarsening process, congestion-driven graph-based Steiner tree construction during the initial routing and multi-iteration refinement with the congestion history to improve the quality of the multilevel routing algorithm[7]. However, these proposed multilevel routing[6-8] mainly focus on global routing. Recently, Lin and Chang[9] proposed a novel framework, named MR, for multilevel full-chip routing.

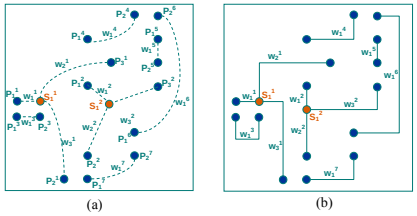
Based on a probabilistic congestion estimation model and two kinds of timing-constrained routing flexibilities: movable Steiner-points and wire detours, Yan and Lin[10] proposed a timing-constrained congestion-driven global routing(TCGR) approach to obtain an initial timing-constrained congestion-driven global routing result without destroying the timing constraint of any routing net. Finally, a simulated-annealing-based timing-constrained rip-up-and-reroute (STRR) approach is proposed to guarantee to obtain better timing-constrained global routing result. However, the flat approach must take much computation time to obtain a timing-constrained global routing result for any larger routing design.

In this paper, according to the result of a set of timing-driven routing trees for all the interconnection nets, a new multi-level timing-constrained full-chip routing (MTFR) in a dynamic hierarchical quad-grid model is proposed to complete full-chip routing in reasonable time. In the proposed MTFR, the global paths of all the routing trees can be assigned by using the timing–constrained flexibility of Steiner points in a top-down timing-constrained congestion-driven global routing (TCGR) process in a quad-grid model. If the global path of any routing tree is successfully not assigned in a top-down TCGR process, a simulated-annealing-based timing-constrained rip-up-and-reroute(STRR) process in a hierarchical quad-grid model will improve the routing tree to complete the global paths of the overflow wires in a hierarchical quad-grid model. Furthermore, all the global paths of the routing trees in a routing plane can be physically assigned by using pattern routing in a bottom-up timing-constrained pattern-driven detailed routing (TPDR) process in a quad-grid model. If the physical path of any routing tree is successfully not assigned in a bottom-up TPDR process, a timing-constrained maze routing(TMR) process in a routing plane will improve the routing tree to assign the timing-constrained physical paths of the failed wires in a uniform grid model. Finally, the experimental results show that the proposed MTFR

approach obtains 100% timing-constrained routing results for all the tested benchmark circuits.

2. Problem Formulation

Basically, the success of any complicated chip seriously depends on the routability of a used full-chip routing methodology. In general, full-chip routing is divided into full-chip global routing and full-chip detailed routing. It is known that a final full-chip global routing result can be obtained by collecting the routing results of a given set of routing trees in a routing plane. In order to maintain the routability of the full-chip routing methodology, full-chip global routing must focus on the congestion control among all the routing trees. Hence, full-chip global routing must be treated as congestion-driven global routing in a routing plane. Based on a full-chip global routing result and physical design rules in a routing plane, full-chip detailed routing must focus on the physical assignment of global paths for all the routing trees. If the timing constraint of any routing tree is maintained in full-chip global and detailed routing, the full-chip routing process can be further treated as a timing-constrained full-chip routing process. Hence, the timing-constrained full-chip routing(TFR) problem can be formulated as follows: Given a set of routing trees, $T = \{T_1, T_2,\ldots, T_N\}$, in a routing plane, each routing tree, T_i, is described by a set of pins, $P_i = \{ p_1^i,\ldots, p_{\delta(i)}^i \}$, a set of Steiner-points, $S_i = \{ S_1^i,\ldots, S_{\theta(i)}^i \}$ and a set of wires, $W_i = \{ w_1^i,\ldots, w_{\varepsilon(i)}^i \}$, where $\delta(i)$ represents the number of given pins, $\theta(i)$ represents the number of generated Steiner-points and $\varepsilon(i)$ represents the number of connecting wires in T_i, respectively. Based on the timing-constrained routing flexibilities of all the routing trees, the timing-constrained full-chip routing (TFR) problem is to reassign the Steiner-points of all the routing trees in T onto feasible timing-constrained positions and further assign the physical paths of all the routing trees without destroying the required timing constraints in T.



(a) (b)

Fig. 1 Timing-constrained full-chip routing

As shown in Fig. 1(a), given a set of seven routing trees, $T = \{T_1, T_2,\ldots, T_7\}$, with a generated Steiner-point, S_1^1, in T_1 and a generated Steiner-point, S_1^2, in T_2. The TFR problem is to reassign two Steiner-points, S_1^1 and S_1^2, onto feasible timing-constrained positions and further assign the physical paths of all the routing trees without destroying the timing constraints in T as shown in Fig. 1(b).

3. Hierarchical Quad-grid Model

In order to understand the routability of all the routing nets in a routing plane, a routing plane can be partitioned into a set of uniform grid cells in a static uniform-grid model and the routability of any grid cell can be estimated by the congestion of the grid cell. If any grid cell in a routing plane has 100% routability, all the routing nets may be fully routed in a routing plane. In a static uniform-grid model, a routing plane can be divided into a set of mxn uniform grid cells, $\{G_{1,1}, G_{1,2}, \ldots G_{1,n},\ldots G_{m,1},\ldots G_{m,n}\}$, where m is the number of horizontal grid cells and n is the number of vertical grid cells in a routing plane, respectively. For any grid cell, $G_{i,j}$, there exist a horizontal grid edge, $H_{i,j}$, on the bottom boundary, and a vertical grid edge, $V_{i,j}$, on the right boundary. Hence, there exist (m-1)xn horizontal grid edges, $\{H_{1,1}, H_{1,2}, \ldots H_{1,n},\ldots H_{(m-1),1},\ldots H_{(m-1),n}\}$ and mx(n-1) vertical grid edges, $\{V_{1,1}, V_{1,2}, \ldots V_{1,(n-1)},\ldots V_{m,1},\ldots V_{m,(n-1)}\}$ in a routing plane.

To our knowledge, many congestion-driven global routing approaches have estimated the congestion of any grid cell in a static uniform-grid model. For a given set of routing trees in a routing plane, it is assumed that all the given pins and its related Steiner-points in any routing tree are fixed. Any routing tree can be treated as a set of two-endpoint wires in a routing plane. In a static uniform-grid model, some two-endpoint wires may be fully contained inside a grid cells if the grid size is too large, and the ignoring result of the inward wires will make the congestion estimation in some grid cells to be under-estimated. On the other hand, many empty grid cells may be yielded in a routing plane if the grid size is too small, and the number of possible paths of any routing wire will make the congestion estimation of some grid cells be over-estimated. Hence, the uniform grid size in a static uniform-grid model will lead to the routability estimation to be less accurate and more time-consuming.

To accurately estimate the congestion of all the routing trees in a routing plane, an effective dynamic quad-grid model has been proposed by Yan et al.[11] and used to estimate the routability for congestion-driven global routing. Basically, a dynamic hierarchical quad-grid model is formulated as follows: Initially, a routing plane can be treated as a single grid cell, G. As any wire in a routing tree is fully contained inside a grid cell, the corresponding grid cell will be geometrically divided into four equal-sized grid subcells and the index of the divided grid subcells is numbered as that of four quadrants in a geometrical plane. For example, the grid cell, G, is

divided into four grid subcells, G_1, G_2, G_3 and G_4, and the grid cell, G_1, can be further divided into four grid subcells, G_{11}, G_{12}, G_{13} and G_{14}, etc.. The recursive division of a original routing plane cannot stop until any wire in a routing tree is not fully contained inside a grid cell. Hence, the hierarchical division of the grid cells in a routing plane can be modeled as a quad tree and any leaf node in the corresponding quad tree can represent a final resultant grid cell in a dynamic hierarchical quad-grid model.

Refer to the routing trees in Fig. 1, the hierarchical division of the grid cells in a routing plane is illustrated in Fig. 2(a) and its corresponding quad tree is shown in Fig. 2(b). Clearly, all the leaf nodes in the quad tree represent the final resultant grid cells in a dynamic hierarchical quad-grid model.

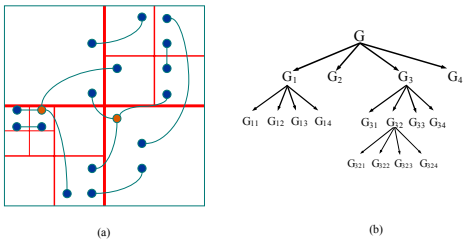


Fig. 2 Hierarchical quad-grid model and its quad-tree

In a dynamic hierarchical quad-grid model, the adjacent relation between two resultant grid cells can be defined as follows:

Definition 1: Given two grid cells, $G_{\alpha(1)\alpha(2)...\alpha(m)}$ and $G_{\beta(1)\beta(2)...\beta(n)}$, $1 \le \alpha(i) \le 4$, $1 \le i \le m$, and $1 \le \beta(i) \le 4$, $1 \le i \le n$, $G_{\alpha(1)\alpha(2)...\alpha(m)}$ is horizontally(vertically) adjacent to $G_{\beta(1)\beta(2)...\beta(n)}$ if one of the following conditions is satisfied as

(1) $\alpha(i) = \beta(i)$, $1 \le i \le m-1$, and $(\alpha(i), \beta(i)) = (2, 1)$ or $(3, 4)$[$(3, 2)$ or $(4, 1)$], if $m = n$, or

(2) $G_{\alpha(1)\alpha(2)...\alpha(m)}$ is horizontally(vertically) adjacent to $G_{\beta(1)\beta(2)...\beta(n)}$ and $(\alpha(i), \beta(i)) = (1, 2)$ or $(4, 3)$[$(1, 4)$ or $(2, 3)$], if $m = n$, or

(3) $G_{\alpha(1)\alpha(2)...\alpha(m)}$ is horizontally(vertically) adjacent to $G_{\beta(1)\beta(2)...\beta(n)}$ and $\alpha(i) = 1$ or 4(1 or 2), $n+1 \le i \le m$, if $m > n$, or

(4) $G_{\alpha(1)\alpha(2)...\alpha(m)}$ is horizontally(vertically) adjacent to $G_{\beta(1)\beta(2)...\beta(n)}$ and $\beta(i)=2$ or 3(3 or 4), $m+1 \le i \le n$, if $m < n$.

In general, the congestion condition on any grid edge seriously depends on the pin distribution and the path assignment of all the routing trees in a routing plane. In [11], a probabilistic estimation in a hierarchical quad-grid model is proposed to predict the congestion of the routing trees in a routing plane as follows: For any two-endpoint wire, *w*, in a hierarchical quad-grid model, all the overlapping routing grids for the wire, *w*, can be further obtained by finding all the grid cells overlapping the minimum routing region. According to the adjacent relation of the overlapping routing grid cells, in a hierarchical quad-grid model, an adjacent graph, G_w, can be constructed by assigning the routing grid cells as vertices and all the adjacent relations as edges. Basically, two grid cells including two endpoints in the wire, *w*, are treated as a start vertex and a target vertex in G_w, and the direction of the adjacent edges in G_w is assigned according to the concept of minimum routing distance from the start vertex to the target vertex. Clearly, an adjacent graph for any two-endpoint wire is acyclic. Initially, the number of possible paths on the start vertex is assigned as 1. For any vertex, *v*, in G_w, the number of possible paths can be obtained by adding all the possible paths on the in-degree edges of the vertex *v* and broadcasted to the out-degree edges of the vertex *v*. Finally, the number of possible paths on the target vertex can be easily obtained and the number, *N(w)*, of possible global paths for the wire, *w*, can be defined as the number of possible paths on the target vertex.

For any two-endpoint wire, *w*, with two endpoints located inside G_s and G_t, the wire through the grid edge, $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, between the grid cells, $G_{\alpha(1)\alpha(2)...\alpha(m)}$ and $G_{\beta(1)\beta(2)...\beta(n)}$, can be divided into the first wire, w_1, from G_s to $G_{\alpha(1)\alpha(2)...\alpha(m)}$, the second wire, w_2, from $G_{\alpha(1)\alpha(2)...\alpha(m)}$ to $G_{\beta(1)\beta(2)...\beta(n)}$, and the third wire, w_3, from $G_{\beta(1)\beta(2)...\beta(n)}$ to G_t. Hence, the number, $N_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}(w)$, of possible global paths through $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, can be obtained as $N(w_1)x\ N(w_2)x\ N(w_3)$. Since $N(w_2) = 1$, $N_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}(w)$ is equal to $N(w_1)xN(w_3)$. Furthermore, it is assumed that the routing probability of any possible path is equal, the probability, $P_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}(w)$, of the global paths through $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, can be obtained as

$$P_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}(w)=\frac{N(w_1)xN(w_3)}{N(w)}$$

Given a set of two-endpoint wires, $\{w_1, w_2, \ldots, w_k\}$ in a hierarchical quad-grid model, and the routing capacity, $Cap_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, on the grid edge, $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, the probabilistic congestion, $C_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ on the grid edge, $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, can be further defined as

$$C_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)} = \frac{\sum_{i=1}^{k} P_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}(w_i)}{Cap_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}}$$

4. Multilevel Timing-Constrained Full-Chip Routing

Because of a large number of routing nets in a modern complicated chip, the TFR problem is always solved by a timing-consuming flat approach. To complete a full-chip routing process in a reasonable time, the multilevel approach can be proposed to reduce the computational time for the TFR problem. The proposed multilevel timing-constrained full-chip routing(MTFR) approach is divide into four phases: (1) Top-down timing-constrained congestion-driven global routing(TCGR) in a quad-grid model, (2) Simulated-annealing-based timing-constrained rip-up-and-reroute (STRR) improvement in a hierarchical quad-grid model, (3) Bottom-up timing-constrained pattern-driven detailed routing(TPDR) in a quad-grid model and (4) Timing-constrained maze routing(TMR) improvement in a uniform grid model. As illustrated in Fig. 3, given a set of routing trees with their timing constraints in a routing plane, the global paths of all the routing trees can be assigned by using the timing–constrained flexibility of Steiner points in a top-down TCGR process. If the global path of any routing tree is successfully not assigned, a STRR improvement process will help the overflow routing tree to complete the assignment of its global path in a hierarchical quad-grid model. Furthermore, all the global paths of the routing trees in a routing plane can be physically assigned by using the technique of pattern routing in a bottom-up TPDR process. If the physical path of any routing tree is physically not assigned, a TMR improvement process in a routing plane will help the failed routing tree to assign its timing-constrained physical path in a uniform grid model.

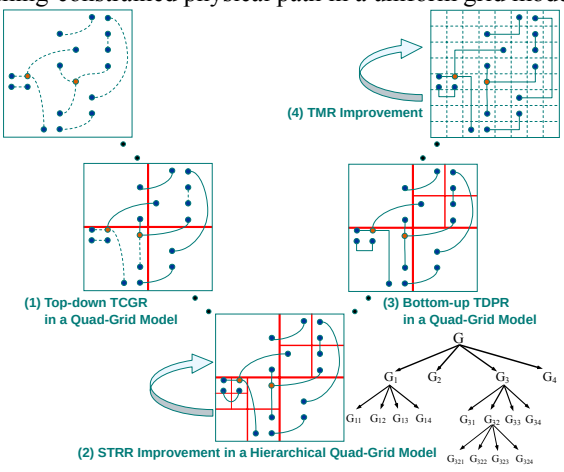


Fig. 3 Hierarchical timing-constrained full-chip routing

4.1. Top-Down TCGR in Quad-Grid Model

Given a set of routing trees with their timing constraints in a routing plane, first, it is assumed that all the Steiner-points in the routing trees are located on fixed positions and any routing tree is decomposed into some two-endpoint wires. Basically, the original routing plane can be treated as an initial grid cell. If any two-endpoint wire is fully contained in the initial grid cell, the grid cell will be geometrically partitioned into four quadrants as four grid subcells by using a "+"-type partitioning line. Basically, two-endpoint wires across the "+"-type partitioning line are further divided into straight wires and diagonal wires. Two endpoints in any straight wire are located in two adjacent quadrants, and two endpoints in any diagonal wire are located in two diagonal quadrants. According to the lengths of four horizontal and vertical grid edges in the "+"-type partitioning line and the width of a routing pitch in the design rule, the horizontal and vertical routing capacities, Cap_H and Cap_V, between two adjacent grid subcells can be obtained by estimating the number of pitches on the horizontal and vertical grid edges. Clearly, there is only one possible global path for any straight wire, and there are only two global paths for any diagonal wire. Hence, the routing probability of any straight wire through its horizontal or vertical grid edge can be obtained as 1, and the routing probability of any diagonal wire through its horizontal or vertical grid edge can be obtained as 0.5. Furthermore, the probabilistic congestion of any horizontal(or vertical) grid edge in the "+"-type partitioning line can be estimated by computing the ratio of the sum of total routing probabilities on any horizontal(or vertical) grid edge and the horizontal(or vertical) routing capacity, Cap_H (or Cap_V). In Fig. 4(a), the straight wires and the diagonal wires in a quad-grid model are shown. Refer to the routing trees in Fig. 1, it is assumed that the horizontal or vertical routing capacity, Cap_H or Cap_V, is 4, the probabilistic congestion of two horizontal grid edges in the "+"-type partitioning line are 0.25 and 0.75, and the probabilistic congestion of two vertical grid edges in the "+"-type partitioning line are 0.5 and 0.75 as shown in Fig. 4(b).

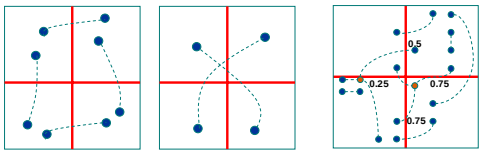


Fig. 4 Probabilistic congestion estimation in a quad-grid model

Second, it is assumed that all the Steiner-points in any routing tree in a routing plane are movable and the structure of any Steiner-point and its related wires constructs a Y-type wire. According to the discussion of the timing-constrained location flexibility of Steiner-point in any Y-type wire[10], the Steiner-point in any Y-type wire has its TSLR in the routing plane. Without destroying the timing constraints of all the routing trees,

any Steiner-point in a routing tree can be located on the feasible position inside its TSLR to balance the local congestion of all the horizontal and vertical grid edges in the TSLR. Intuitively, if any Steiner-point is located on the feasible position, the local congestion balance in all TSLRs will lead to the global congestion balance in a routing plane. Refer to the probabilistic congestion of four horizontal and vertical grid edges in Fig. 4, the Steiner-point movement in a routing tree can reduce the maximal probabilistic congestion of four horizontal and vertical grid edges from 0.75 to 0.5 as shown in Fig. 5.

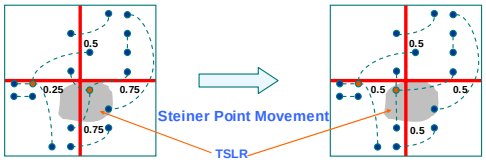


Fig. 5 Timing-constrained Steiner-point Assignment in a quad-grid model

Finally, it is again assumed that all the Steiner-points in any routing tree are fixed, and two-endpoint wires in any routing tree are divided into straight wires and diagonal wires. Since there is only one possible global path for any straight wire, the global paths of all the straight wires can be directly assigned through their horizontal or vertical grid edges. After assigning the global paths of all the straight wires, the probabilistic congestions and the routing capacities on four horizontal and vertical grid edges must be further modified. Since there are only two possible global paths for any diagonal wire, the global path of any diagonal wire can be assigned by selecting a global path with minimal probabilistic congestion. After assigning the global path of any diagonal wire, the probabilistic congestions and the routing capacities on four horizontal and vertical grid edges must be dynamically modified. Until the global paths of all the diagonal wires are assigned, the global assignment of the routing trees across the "+"-type partitioning line will stop. Refer to the routing trees in Fig. 5, the assignment of the global paths of all the straight wires and all the diagonal wires is shown in Fig. 6(a) and Fig. 6(b), respectively.

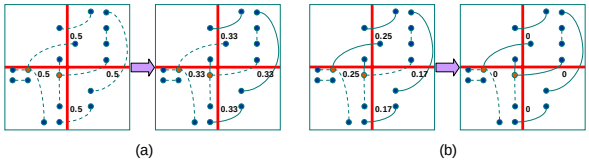


Fig. 6 Assignment of the global paths of the routing trees in a quad-grid model

Given a set of routing trees with their timing constraints in a routing plane, if any two-endpoint wire is fully contained in the routing plane, the routing plane will be geometrically partitioned into four quadrants as four routing subplanes by using a "+"-type partitioning line. Furthermore, the global paths of all the two-endpoint wires across the "+"-type partitioning line can be assigned by running a TCGR process in a quad-grid model. Similarly, the global paths of all the two-endpoint wires inside four routing subplanes can be recursively assigned by running a TCGR process in a quad-grid model. Hence, the global paths of the routing trees with their timing constraints in a routing plane can be assigned by using a top-down TCGR algorithm, *TopDown_TCGR*, in a quad-grid model and the *TopDown_TCGR* algorithm can be described as follows:

TopDown_TCGR (T, G)
Input: a set of routing trees, T, with their timing constraints in a routing plane, G;
{ Partition the grid cell G into four uniform subgrids G_1, G_2, G_3 and G_4;
Compute the probabilistic congestions of all the grid edges for T in a quad-grid model;
Assign the Steiner points of the routing trees in T onto feasible positions inside the related TSLRs;
Assign the congestion-driven global paths of all the straight and diagonal wires on the grid edges;
for (i=1;i<=4;i++)
if (any wire in T is fully contained in G_i)
TopDown_TCGR(T, G_i);
Output the assignment of the global paths of all the routing trees in T;
}

4.2. STRR Improvement in Hierarchical Quad-Grid Model

As the global paths of all the routing trees in a routing plane are assigned by running a top-down TCGR process in a quad-grid model, the recursive partition of a routing plane in a quad-grid model will form the concept of a routing plane in a hierarchical quad-grid model. Given the routing capacity, $Cap_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, on the grid edge, $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, between $G_{\alpha(1)\alpha(2)...\alpha(m)}$ and $G_{\beta(1)\beta(2)...\beta(n)}$ in a hierarchical quad-grid model, the number of wires across the grid edge, $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, is designated as the edge demand, $d_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, of the grid edge, $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$. Hence, the edge overflow, $Overflow_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, of the grid edge, $H_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$ or $V_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$, is defined as

$$Overflow_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)} = \begin{cases} d_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)} - Cap_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}, \\ \quad if \quad d_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)} - Cap_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)} > 0, \\ 0, \quad if \quad d_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)} - Cap_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)} \le 0, \end{cases}$$

Furthermore, the global overflow, $Overflow$, in a routing plane in a hierarchical quad-grid model is defined as the summation of the edge overflows of all the grid edges as

$$Overflow = \sum_{for\ \ all\ grid\ edges} Overflow_{\alpha(1)\alpha(2)...\alpha(m),\beta(1)\beta(2)...\beta(n)}$$

If the global overflow, $Overflow$, is equal to 0 in a routing plane in a hierarchical quad-grid model, the assignment of the global paths of all the routing trees in a routing plane will be completed by using a top-down TCGR process in a quad-grid model. In contrast, if the edge overflow of any grid edge is larger than 0, an STRR improvement process in a hierarchical quad-grid model must be proposed to release the overflow condition of any grid edge by reassigning the global paths of all the overflow wires in a routing plane.

Based on the timing-constrained location flexibility of Steiner-point in any Y-type wire and the timing-constrained detour flexibility in any two-endpoint wire, three timing-constrained rip-up-and-reroute operations, *Steiner-point assignment-and-reroute*, *Steiner-point deletion-and-reroute* and *Detour assignment-and-reroute*, in the STRR improvement approach[10] are proposed to release the overflow conditions of all the grid edges in a hierarchical quad-grid model. According to the estimation of the probabilistic congestion in a hierarchical quad-grid model[11], the probabilistic congestions of all the horizontal and vertical grid edges and the horizontal and vertical routing capacities in a hierarchical quad-grid model must be computed. Furthermore, the ripped wires after running any timing-constrained rip-up-and-reroute operation can be rerouted by running the assignment of global paths in a hierarchical quad-grid model. For the assignment of the global path of any ripped wire in a hierarchical quad-grid model, any grid cell in the timing-constrained routing region of the ripped wire is treated as a vertex, the adjacent relation between two adjacent grid cells in the timing-constrained routing region of the ripped wire as an edge, and the probabilistic congestion on the horizontal or vertical grid edge as an edge weight to construct a path-connection graph. Furthermore, the shortest path between two endpoints in the ripped wire in the path-connection graph can be found and the global path of the ripped wire can be assigned by following the connecting vertices in the resultant shortest path. After assigning the global path of any ripped wire, the probabilistic congestions of the horizontal and vertical grid edges and the horizontal and vertical capacities in a hierarchical quad-grid model must be dynamically modified. Hence, the global paths of all the ripped wires can be assigned to release all the overflow conditions in a hierarchical quad-grid model.

Refer to the global paths of the routing trees in Fig. 6. Clearly, there exist an edge overflow between two grid cells, G_{321} and G_{322}, in Fig. 7(a). By using the *Detour assignment-and-reroute* operation in a hierarchical quad-grid model, the global path of the ripped wire is reassigned and the final global routing result is obtained by running a STRR improvement process in a hierarchical quad-grid model and shown in Fig. 7(b).

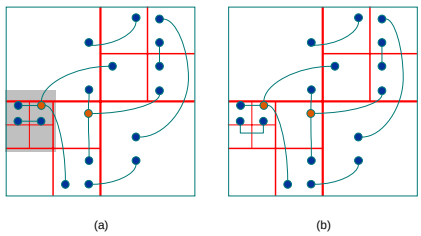


Fig. 7 STRR improvement in a hierarchical quad-grid model

In the proposed STRR improvement in a hierarchical quad-grid model, the set of all the timing-constrained global routing results is defined as the solution space and the global overflow, $Overflow$, of the timing-constrained global routing result is defined as the cost function. By using three timing-constrained rip-up-and-reroute operations, *Steiner-point reassignment-and-reroute*, *Steiner-point deletion-and-reroute* for the Steiner point in any Y-type wire and *Detour assignment-and-reroute* in any S-type wire, to perturb the solution space in a hierarchical quad-grid model, the overflow conditions of all the grid edges in a hierarchical quad-grid model will be gradually released. As the global overflow, $Overflow$, of any timing-constrained global routing result is equal to 0 or the simulated temperature is converged, a final timing-constrained global routing result will be obtained.

4.3. Bottom-Up TPDR in Quad-Grid Model

Since the grid cells in a hierarchical quad-grid model are represented as a quad tree, the routing result in the four adjacent grid cells partitioned by a "+"-type line can be further integrated into the routing result in a larger grid cell in a bottom-up manner. Until the final integrated grid cell corresponds to a full routing plane, the routing result in the final grid cell will be the final routing result in a routing plane. In integrating four adjacent grid cells into a larger grid cell in a bottom-up manner, the partial or full global paths of all the routing trees in the four adjacent grid cells must be physically assigned onto the horizontal and vertical grid edges and routed as special wiring patterns by running a bottom-up TPDR process in a quad-grid model. If all the grid cells in a quad tree are integrated into a full routing plane and the global paths of all the routing trees are physically assigned by using special routing patterns, a final detailed routing result will be obtained.

In a bottom-up TPDR process in a quad-grid model, it is assumed that all the Steiner-points in the routing trees are located on fixed positions and any routing tree is decomposed into some two-endpoint wires. First, any two-endpoint wire across a "+"-type partitioning line is represented as a straight line connecting two endpoints.

Basically, one crossing point on the "+"-type partitioning line can be obtained for any straight wire and two crossing points on the "+"-type partitioning line can be obtained for any diagonal wire. For two-endpoint wires whose global paths cross the same grid edge, the physical ordering of these wires can be further decided according to the positions of their crossing points on the grid edge. By finding the position of the nearest track of its crossing point on the grid edge, the position of the boundary pin of any wire on the same grid edge can be obtained. According to the positions of the boundary pins on the "+"-type partitioning line, any straight wire can be divided into two smaller wires and any diagonal wire can be divided into three smaller wires. Clearly, these divided smaller wires are fully contained in four grid cells in a quad-grid model. The smaller wires inside any of four grid cells can be physically routed by using an I-type, L-type or Z-type path in a pattern routing process. Hence, the physical path of any two-endpoint wire can be assigned by using the connection of I-type, L-type or Z-type paths in a bottom-up pattern-driven detailed routing process.

Refer to the global routing result in Fig. 7, if the physical paths of all the two-endpoint wires in the third level have been assigned, the representation of a three-level quad tree has been reduced into that of a two-level quad tree in a hierarchical quad-grid model. For all the two-endpoint wires in the second level, the crossing points of the wires are marked as "X" in Fig. 8(a), the assignment of the boundary pins of the wires is shown in Fig. 8(b), and the pattern-driven detailed routing result for a bottom-up TPDR process in a quad-grid model is shown in Fig. 8(c).

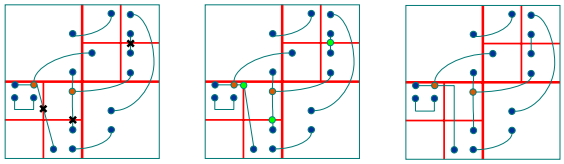


Fig. 8 Boundary pin assignment and detailed pattern routing in a quad-grid model

Furthermore, there may be some redundant bends in the physical path of any two-endpoint wire. In general, redundant bends in the connection of any physical path will yield redundant vias to increase the timing delay. Hence, the reduction of bends in the physical path of any two-endpoint wire is applied to rebuild the connection of the physical path with minimal vias. Furthermore, the detailed routing result inside four grid cells is integrated into the detailed routing result inside a larger grid cell by deleting the "+"-type partitioning line.

Refer to the detailed routing result in the second level, the physical paths of all the two-endpoint wires in the second level can be integrated into the routing result in the first level, and the representation of a two-level quad tree is reduced into that of a one-level quad tree in a hierarchical quad-grid model. In Fig. 9, the physical paths of all the two-endpoint wires in the first level are assigned by assigning boundary pins, using pattern-driven detailed routing and reducing redundant bends. However, there still exists a two-endpoint wire whose physical path is not assigned after running a bottom-up TPDR process in a quad-grid model.

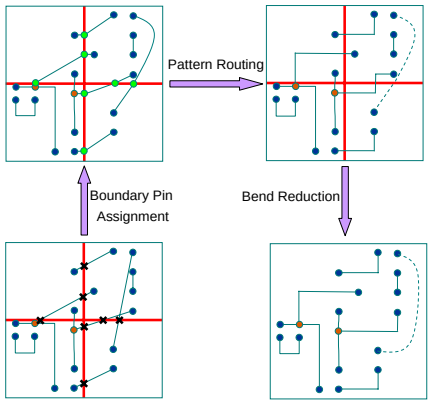


Fig. 9 Bottom-up TPDR in a quad-grid model

Given the global paths of all the routing trees in a hierarchical quad-grid model, the physical paths of all the two-endpoint wires across all the "+"-type partitioning lines can be recursively assigned by running a bottom-up TPDR process in a quad-grid model and the bottom-up TPDR algorithm, *BottomUp_PTDR*, can be described as follows:

BottomUp_TPDR (T, G)
Input: the global paths of the routing trees, T, in a hierarchical quad-grid model;
{ Assign the boundary pins of all the two-endpoint wires in the lowest level of G;
Assign the physical paths of the wires in the lowest level of G by using pattern-driven detailed routing;
Reduce the redundant bends of the physical paths of all the two-endpoint wires in the lowest level of G;
Combine the detailed routing results in the lowest level of G and Modify G;
If (G is not a single grid cell)
BottomUp_TPDR(T, G);
Output the physical paths of all the routing trees, T;
}

4.4. TMR Improvement in Uniform Grid Model

As the physical paths of the routing trees in a hierarchical quad-grid model are assigned by running a bottom-up TPDR process in a quad-grid model, the routing result in a hierarchical quad-grid model will be physically assigned and integrated into the routing result in a final routing plane according to the recursive

integration of the grid cells in a hierarchical quad-grid model. Based on the size of a routing pitch, the final routing plane can be further divided into uniform grid cells in a uniform grid model. As the physical path of any two-endpoint wire across a "+"-type partitioning line is not assigned by running a bottom-up TPDR process in a quad-grid model, the timing-constrained region of the unassigned wire in a uniform grid model will be found. Furthermore, the physical path of the unassigned wire inside its timing-constrained region can be assigned by running a TMR improvement process in a uniform grid model. Refer to the detailed routing result in Fig. 9, it is clear that there exists a two-endpoint wire whose physical path is not assigned after running a bottom-up TPDR process in a quad-grid model. In Fig. 10, the timing-constrained region of the unassigned wire can be found and the physical path of the unassigned wire can be further assigned by running a TMR improvement process in a uniform grid model.

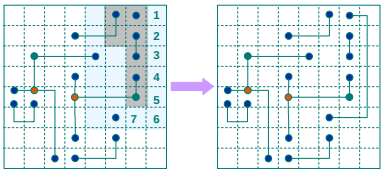


Fig. 10 TMR improvement in a uniform grid model

5. Experimental Results

The proposed MTFR approach has been implemented by using standard C++ language and run on a Pentium IV 2.8G machine with 512M memory. The first set of ten benchmark circuits, a9c3, ac3, ami33, ami49, apte, hc7, hp, playout, xc5 and xerox, and the second set of six benchmark circuits, S5378, S9234, S13207, S15850, S38417 and S38584 are applied to test the proposed MTFR approach.

Table I shows the experimental results of the MTFR approach for the first benchmark circuit set with timing constraints. Basically, all of the routing trees and their timing constraints for any benchmark circuit are from the tested circuits in [10]. In this table, "Time1" and "Time2" denote the CPU time of the timing-constrained global and full-chip routing results, respectively. The experimental results show that the MTFR approach is faster than our previous approach[10] to obtain 100% timing-constrained routing results.

Table II shows the experimental results of the MTFR approach for the second benchmark circuit set without any timing constraint. The routing trees in all the tested circuits can be constructed by using a Steiner tree algorithm[12]. Basically, the MR[9] approach is based on a static uniform grid model. The experimental results show that the MTFR approach in a dynamic quad-grid model can use less CPU time to obtain 100% timing-constrained routing results than the MR approach.

Table I Experimental Results for 1st Circuit Set

circuit	#layers	#Nets	TCGR+STRR[10]		MTFR		
			#Rtd. Nets	Time(s)	#Rtd. Nets	Time1(s)	Time2(s)
a9c3	2	1148	1148(100%)	14.7	1148(100%)	3.7	4.5
ac3	2	200	200(100%)	4.5	200(100%)	2.6	3.4
ami33	2	112	112(100%)	3.4	112(100%)	1.2	1.6
ami49	2	368	368(100%)	6.9	368(100%)	2.2	3.0
apte	2	77	77(100%)	1.4	77(100%)	1.1	1.8
hc7	2	430	430(100%)	12.5	430(100%)	2.8	3.9
hp	2	68	68(100%)	2.0	68(100%)	1.0	1.7
playout	2	1294	1294(100%)	13.6	1294(100%)	3.5	4.1
xc5	2	975	975(100%)	12.7	975(100%)	2.7	3.7
xerox	2	171	171(100%)	4.5	171(100%)	1.3	2.0

Table II Experimental Results for 2nd Circuit Set

circuit	#layers	#Nets	MR[9]		HTFR	
			#Rtd. Nets	CPU(s)	#Rtd. Nets	Time(s)
S5378	3	3124	3124(100%)	11	3124(100%)	9.2
S9234	3	2774	2774(100%)	8	2774(100%)	6.7
S13207	3	6995	6995(100%)	38	6995(100%)	21.3
S15850	3	8321	8321(100%)	58	8321(100%)	35.9
S38417	3	21035	21035(100%)	138	21035(100%)	82.1
S38584	3	28177	28177(100%)	317	28177(100%)	136.4

Reference

[1] C. J. Alpert, J. H. Huang, and A. B. Kahng, "Multilevel circuit partitioning," *IEEE Trans. on Computer-Aided Design*, Vol. 17, pp.655–667, 1998.

[2] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Application in VLSI domain," *IEEE Trans. on VLSI Systems*, Vol. 7, pp.69–79, 1999.

[3] J. Cong, S. Lim and C. Wu, "Performance-driven multilevel and multiway partitioning with retiming," *Design Automation Conference*, pp.274-279, 2000.

[4] T. Chan, J. Cong, T. Kong and J. Shinnerl, "Multilevel optimization for large-scale circuit placement," *International Conference on CAD*, pp.171-176, 2000.

[5] S. C. Lee, J. M. Hsu and Y. W. Chang, "Multilevel large-scale module placement/floorplanning using B*-trees," *the 12th VLSI Dsign/CAD Symposium*, 2001.

[6] Y. L. Lin, Y. C Hsu and F. S Tsai, "Hybrid routing," *IEEE Trans. on Computer-Aided Design*, Vol. 9, pp.151–157, 1990.

[7] J. Cong, J. Fang and Y. Zhang, "Multilevel approach to full-chip gridless routing", *International Conference on CAD*, pp.396-403, 2001.

[8] J. Cong, M. Xie and Y. Zhang, "An enhanced multilevel routing system," *International Conference on Computer-Aided Design*, pp.51-58, 2002.

[9] S. P. Lin and Y. W. Chang, "A novel framework for multilevel routing considering routability and performance," *International Conference on CAD*, pp.44-50, 2002.

[10] J. T. Yan and S. H. Lin, "Timing-constrained congestion-driven global routing," *ASP Design Automation Conference*, pp.683-686, 2004.

[11] J. T. Yan, Y. H. Chen and C. W. Wu, "Probabilistic congestion prediction in hierarchical quad-grid model," *IEEE International Symposium on Circuits and Systems*, pp., 2005.

[12] H. Hou, J. Hu, and S. S. Sapatnekar, "Non-Hanan routing," *IEEE Trans. on Computer-Aided Design*, Vol. 18, pp.436–444, 1999.